Přednáška 9

Obsahem přednášky jsou níže uvedené partie skript [TSA] B. Melichar et al.: Text Search Algorithms, FEE CTU, Prague, 2004, viz webové stránky PAL, oddíl literatura.

Ke zvládnutí látky stačí, aby adept znal jednotlivé automaty a postupy do té míry, aby mohl správně příslušný automat vytvořit nebo příslušný postup provést pro daná konkrétní data. Ukázky takových úloh uvádíme níže. Není nutno znát důkazy jednotlivých faktů, stejně tak není nutno znát zpaměti obecné formulace jednotlivých algoritmů nebo konstrukcí automatů.

[TSA] Kap. 2.2.3.1 Hammingova vzdálenost. Automat vyhledávající v textu všechny podřetězce, které mají od daného vzorku Hammingovu vzdálenost nejvýše *d*.

[TSA] Kap. 2.2.3.2 Levenshteinova vzdálenost. Automat vyhledávající v textu všechny podřetězce, které mají od daného vzorku Levenshteinovu vzdálenost nejvýše *d*.

[TSA] Kap. 2.2.6 Automat vyhledávající v textu všechny podřetězce identické s libovolným vzorkem ze zadané konečné množiny vzorků.

[TSA] Kap. 2.4.1 Deterministická podoba předchozího automatu.

[TSA] Kap. 2.2.6 a Kap. 2.3.3 Automat vyhledávající v textu všechny podřetězce, které vyhovují danému regulárnímu výrazu.

[TSA] Kap. 6.2.2.2 Vyhledávání v textu všech podřetězců, které mají od daného vzorku Hammingovu vzdálenost nejvýše *d* metodou dynamického programování.

[TSA] Kap. 6.2.2.3 Vyhledávání v textu všech podřetězců, které mají od daného vzorku Levenshteinovu vzdálenost nejvýše *d* metodou dynamického programování.

Ukázky úloh

1. Sestavte nedeterministický automat nad abecedou $\{a, b, c\}$, který vyhledává v textu podřetězce mající od vzorku *aacbb* Hammingovu/Levenshteinovu vzdálenost nejvýše 3.

2. Sestavte nedeterministický automat nad abecedou {0, 1}, který vyhledává v textu některý z množiny vzorků {0001, 10, 10101, 1110}. Pro tatáž data sestavte deterministický automat bez použití standardního algoritmu převodu nedeterministického automatu na deterministický.

3. Sestavte nedeterministický automat nad abecedou $\{x, y, z, w\}$, který vyhledává v textu každé slovo vyhovující regulárnímu výrazu $w(x + y(zw + xy)^*)^*$.

4. Sestavte tabulku odpovídající metodě dynamického programování, která v textu *abbabbaccacbbc* nad abecedou {*a*, *b*, *c*, *d*} vyhledává podřetězce mající od vzorku *cddca* Hammingovu/Levenshteinovu vzdálenost nejvýše 2.

Hammingova vzdálenost:

		a	*b*	*b*	*a*	*b*	*b*	*a*	*c*	*c*	*a*	*c*	*b*	*b*	*c*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	3	1	1	1	1	1	1	1	0	0	1	0	1	1	0
d	3	4	2	2	2	2	2	2	2	1	1	2	1	2	2
d	3	4	5	3	3	3	3	3	3	3	2	2	3	2	3
c	3	4	5	6	4	4	4	4	3	3	4	2	3	4	2
a	3	3	5	6	6	5	5	4	5	4	3	5	3	4	5
		1	2	3	4	5	6	7	8	9	10	11	12	13	14

Daný text neobsahuje žádný podřetězec, jehož Hammingova vzdálenost od vzorku *cddca* je nejvýše 2.

		a	*b*	*b*	*a*	*b*	*b*	*a*	*c*	*c*	*a*	*c*	*b*	*b*	*c*
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	1	1	1	1	1	1	1	1	0	0	1	0	1	1	0
d	2	2	2	2	2	2	2	2	1	1	1	1	1	2	1
d	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2
c	4	4	4	4	4	4	4	4	3	2	3	2	3	3	2
a	5	4	5	5	4	5	5	4	4	3	2	3	3	4	3
		1	2	3	4	5	6	7	8	9	10	11	12	13	14

Daný text obsahuje podřetězec *cca* na pozicích 8-10, jehož Levenshteinova vzdálenost od vzorku *cddca* je 2. Neobsahuje žádné další podřetězce, jejichž vzdálenost od vzorku *cddca* je nejvýše 2.